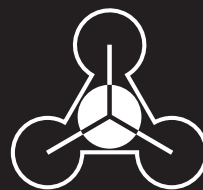




SIMPLEXFM



METACONFORMER  
FIRMWARE

USER  
MANUAL

## CONTENT

GENERAL OVERVIEW .....	3
CONNECTION .....	4
ARCHITECTURE .....	4
SETTINGS CONTROLS .....	13
SAVING SETTINGS .....	16
MIDI CHANNELS .....	17
PLAY MODES .....	18
MONO/STACK .....	18
SUMMARY TABLE OF PARAMETERS .....	20
MASTERING THE INSTRUMENT .....	21
CONCLUSION .....	23
CREDITS .....	24

## GENERAL OVERVIEW

**SimplexFM** is a 4-track digital FM synthesizer for creating deep and brooding soundscapes, industrial noises, sound disasters, and resonant environments that smoothly evolve into one another.

**SimplexFM** runs on Metaconformer hardware. The firmware is already available on the SOMA website.

### Firmware update

1. Download the firmware from the SOMA website
2. Connect your Metaconformer to computer via USB
3. Press and hold the **CFG** button
4. Press the **RST** button. The **CFG** can then be released
5. The Metaconformer must be identified on the computer as a flash drive. All of its LEDs should light up
6. Copy the firmware to the memory of the flash drive
7. Rename the firmware file to **blink.bin**
8. Remove the device safely
9. Restart Metaconformer by pressing **RST**

The firmware update is successful if pressing the **CFG** button turns the green LED on and off.



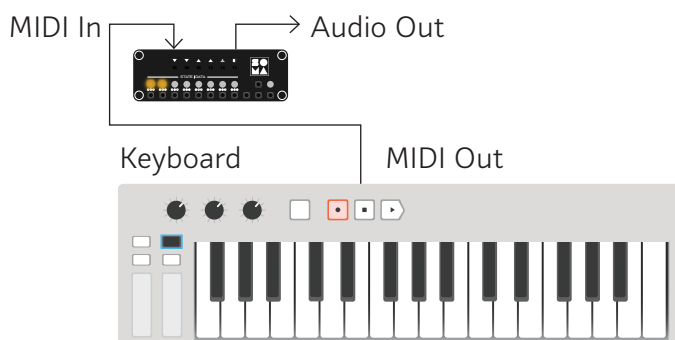
If something went wrong.

1. Delete all files from the flash drive.
2. On a Mac, you need to additionally clear deleted files from the Recycle Bin.
3. Repeat the firmware update procedure.



Sometimes it takes multiple attempts before your computer will recognize your Metaconformer. Allow several seconds before repeat of connection procedure.

## CONNECTION



1. Connect a MIDI keyboard/sequencer to the left MIDI input of the Metaconformer.
2. Pick up the sound from the hybrid output of the Metaconformer.

This is enough to fully control **SimplexFM**.



Right Metaconformer MIDI In is disabled, Metaconformer MIDI Outs work in MIDI Thru mode.

## ARCHITECTURE

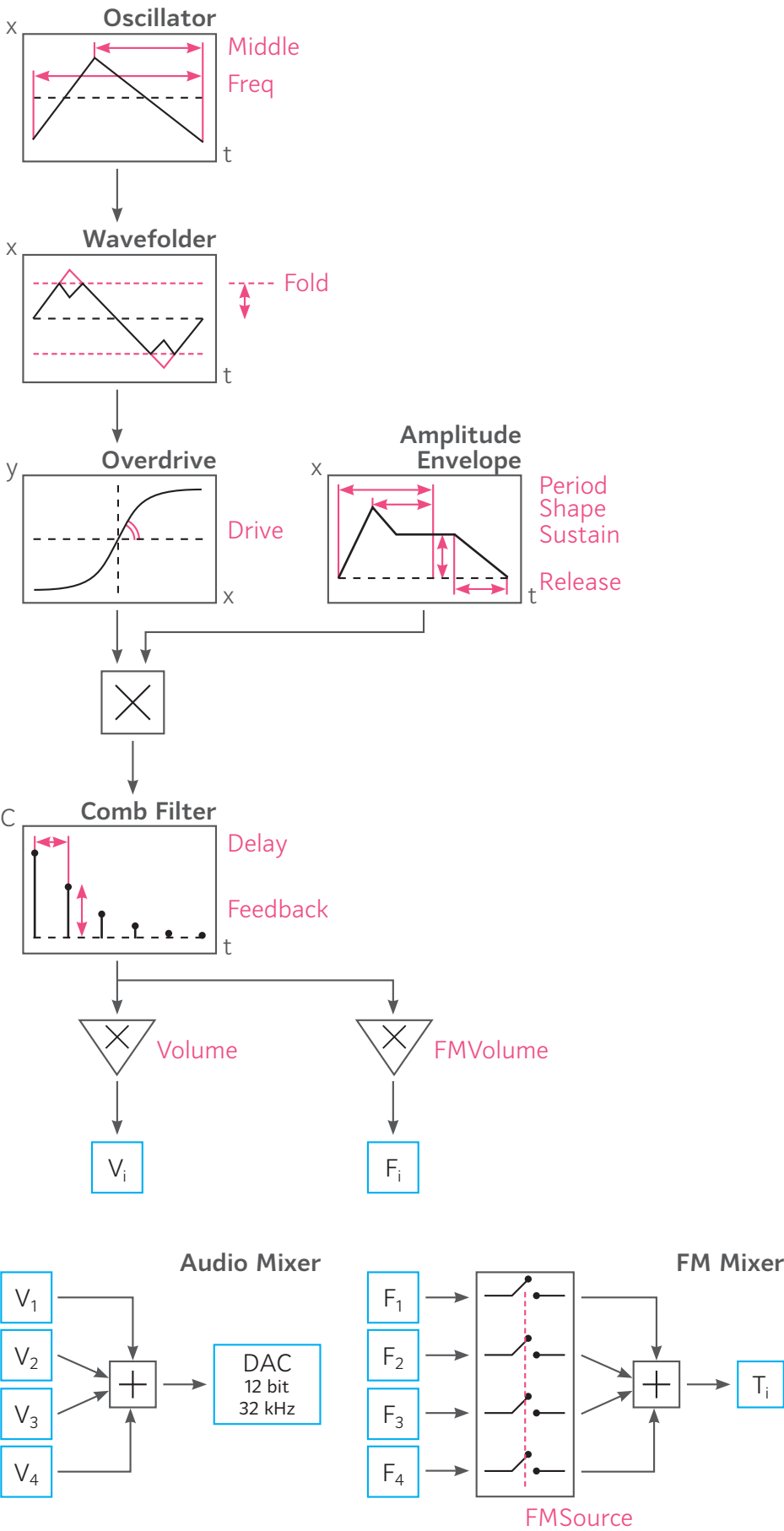
**SimplexFM** is not a polyphonic, but a multitimbral synthesizer. Each of its 4 tracks (voices) is an independent synthesizer with its own settings.

**SimplexFM** is made with an emphasis on additive synthesis. Additional harmonics arise as a result of nonlinear distortions of the signal.

But its unique feature is that tracks can distort each other through frequency modulation.

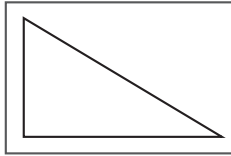
**SimplexFM** is intuitive and does not require the user to have a deep understanding of FM synthesis mechanisms. Instead, it offers a convenient and tactile interface for traveling through the space of sound, where you only need to navigate by your own senses.

The following is a description of the modules that make up **SimplexFM**.

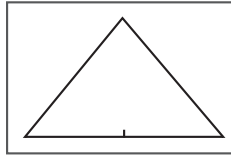


## OSCILLATOR

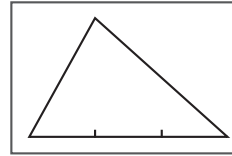
Middle = 0



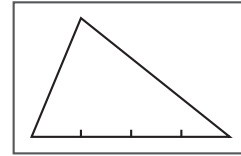
Middle = 1/2



Middle = 1/3



Middle = 1/4

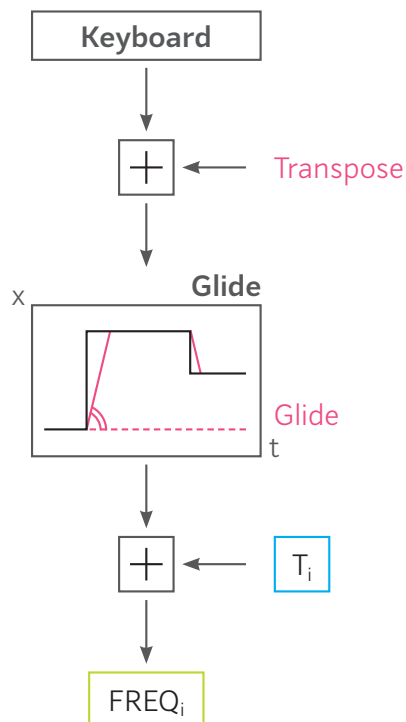


The voice audio path begins with an asymmetric triangle generator. By controlling the asymmetry through the position of the midpoint MIDDLE, the harmonic richness is changed, which is minimal at the triangle and maximal at the saw.

Also, with the oscillator, the user controls the frequency **FREQ**, which depends on the **note** pressed on the keyboard, **TRANPOSE** and the influence of other voices through FM synthesis. More details in **FM Mixer**.

The frequency changing speed is controlled by the **GLIDE** parameter

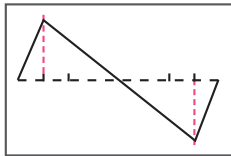
## Frequency Control



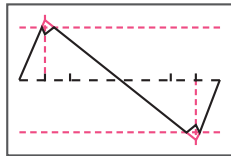
## WAVEFOLDER

The sound of the oscillator enters the wavefolder block, which folds the waveform relative to the offset FOLD level.

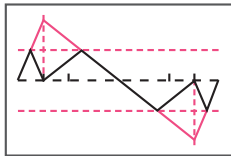
Middle =  $1/4$   
Fold = 1



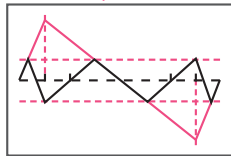
Middle =  $1/4$   
Fold = 0.9



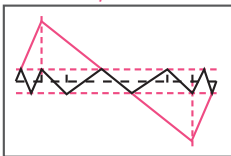
Middle =  $1/4$   
Fold = 0.5




Middle =  $1/4$   
Fold =  $1/3$





Middle =  $1/4$   
Fold =  $1/5$



 After folding, the wave amplitude is normalized.

Such distortion can radically increase the aggressiveness of the timbre and add resonant overtones without using a filter.

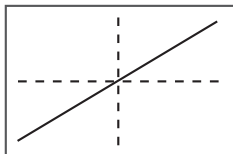
 Due to the fact that the oscillator wave is an asymmetric triangle, two resonant overtones appear in the sound at the output of the wavefolder. The asymmetry of the triangular oscillator controls the interval between resonances that sound like formants.

 With an extreme number of folds, the sound turns into noise, the nature of which depends on the parameters of the oscillator

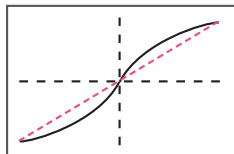
## OVERDRIVE

Increasing of the DRIVE parameter can further saturate the sound coming from the wavefolder with harmonics, up to the complete transformation of the signal into a square.

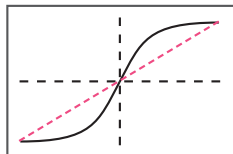
Drive = 0.01



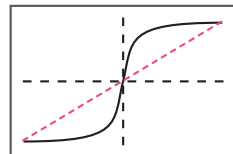
Drive = 1



Drive = 10



Drive = 1000



With slight distortion, it is able to soften the triangle so that it becomes similar to a sine wave, as is done in analog circuits.

## AMPLIFIER

The output of the overdrive block is multiplied by the volume envelope value.

The volume envelope has 4 familiar stages: Attack, Decay, Sustain, Release.

The envelope can operate in cyclic mode. The CYCLE parameter is responsible for this. When this mode is active, the envelope begins to act as a low frequency oscillator, moving through the Attack and Decay stages.

These two stages form an asymmetrical triangle, just like the oscillator, so it was decided to leave the same controls at the envelope.

The PERIOD parameter is similar to the frequency of an asymmetric triangle oscillator. It controls the total time to complete the Attack and Decay stages.

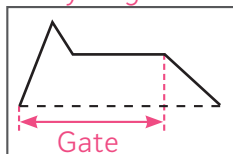
The parameter that determines the position of the midpoint SHAPE is similar to the MIDDLE parameter. It determines the duration of the Attack stage compared to the Decay stage within a period.

If you turn off the cycle mode, the envelope will go through all 4 stages.

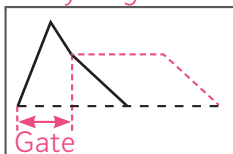


The envelope has both very short parameter values for creating sharp and biting sounds, as well as ring modulation. But also very long, almost imperceptible rise and fall rates for creating slow drones and pads.

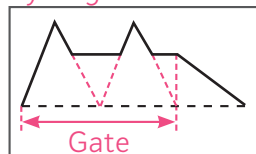
No cycling



No cycling



Cycling enabled



Voice LED 1-4 has a brightness proportional to the voice volume envelope.



## COMB FILTER

Each voice has a comb filter unit through which the signal is passed after its amplification.

The DELAY lengths of the comb filter are adjusted to match the periods of the MIDI notes.

At long delay lengths (at low resonant frequencies of the comb filter) it can be used as an echo.

The decay time controlled by the FEEDBACK parameter. The larger the modulus, the longer the sound takes to decay out.



FEEDBACK has both positive and negative values. The sign of the coefficient determines the frequency response of the filter.



To link the delay length of the comb filter to the value of the voice note, activate KEYTRACK parameter.

## VOLUME

The volume of the voice in the mix is set by the VOLUME parameter. Volume has both positive and negative values.



Negative VOLUME values allow you to subtract one timbre from another. If two sounds have common harmonics, then they can be subtracted from the signal in this way. This works as a very unusual way of filtering.

The sum is fed to a DAC connected to the Metaconformer's hybrid output, which has 12-bit resolution and a 32 kHz sampling rate.

## FREQUENCY MODULATION

In **SimplexFM** the sound of each voice goes to the audio output. But separately, this sound can also be used for frequency modulation.

The FMVOLUME parameter determines the amplitude of the voice signal that can participate in frequency modulation. In other words, FMVOLUME controls the intensity of the voice as a modulator.

This signal goes to the FM Mixer input.

## FM MIXER

The great power of **SimplexFM** lies in the fact that its voices can modulate each other's frequency in any possible combination.

💡 Since SimplexFM voices already have the ability to manipulate harmonic richness through additive parameters, FM synthesis can be perceived as an effect that distorts the waveform in a highly non-linear and interesting way.

Instead of choosing one of the algorithms (a diagram that describes the connection of operators) of frequency modulation, **SimplexFM** allows you to select for each track which tracks are its modulator.

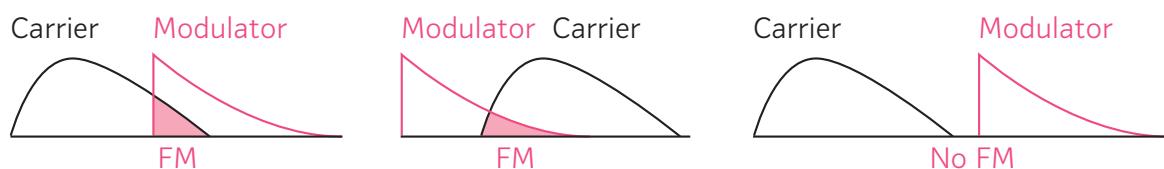
Each track has its own FM mixer, summing the incoming modulator signals. The result of the addition modulates the frequency of the selected voice.

The FMSOURCE parameter determines which voices are modulators of the current one.

👆 In classic FM synthesizers, the FM algorithm is common to all operators. Pressing a note on the keyboard triggers all operators simultaneously.

At **SimplexFM** all operators are **independent** voices.

- If you start the carrier, but do not start its modulator, FM synthesis will not happen.
- If you start the modulator when the carrier is already quiet, FM synthesis will also not happen.
- FM synthesis will only occur when the sounds of the carrier and modulator are overlay.



FMSOURCE is represented as 4 bits. It's easier to say that these are 4 virtual switches. Each switch corresponds to one of the tracks from left to right.

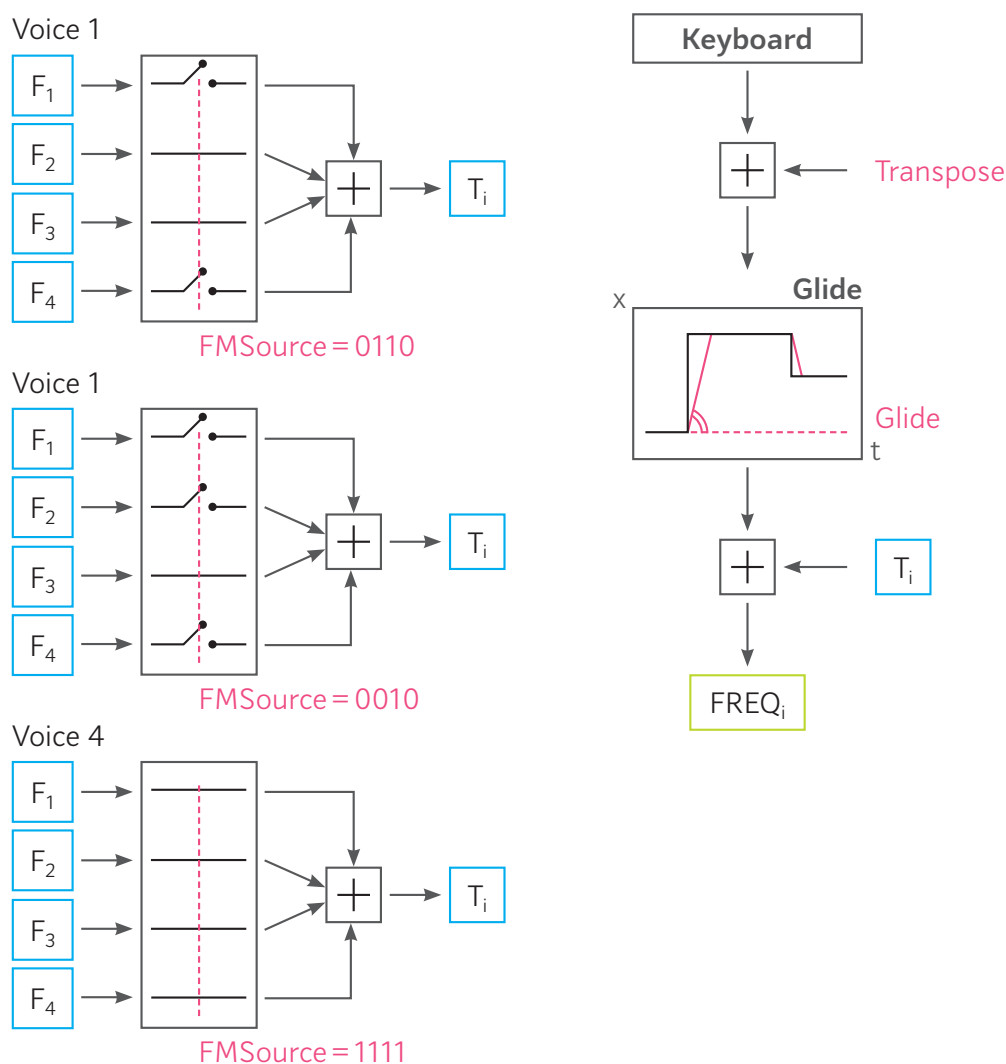
If the LED in the switch position is lit, then the switch is closed.

● = 1

If the LED in the switch position does not light up, then the switch is open.

● = 0

The FM mixer of each voice sums the signals of only those voices whose switch is closed. The result modulates the frequency of the current voice.



Simplex allows you to directly control the connections of voices, choose which voice modulates whom.

There is no limit on the number of connections or the role of voice. It can be a carrier, a modulator, or combine both roles.

💡 If you want the voice to become only a modulator, just set its VOLUME to 0. It will disappear from the mix, but will still be able to affect other tracks.

This approach helps you intuitively move from simple to complex in search of interesting and unexpected sounds, focusing only on your feelings. You connect the voices with each other and listen to the result. You can complicate the synthesis algorithm by adding connections, or you can simplify it by removing them.

💡 Even the simplest algorithms, where one voice modulates another in simplexFM, sound unexpected and fresh.

This is because the comb filter is not just an effect that processes the result of FM synthesis. It also participates in synthesis, accumulating the past, which has an effect on the present and future.

## HARMONICS



The frequencies of harmonics resulting from FM synthesis depend on the ratio of the modulator frequency to the carrier frequency.

To put it simply, the number of synthesized harmonics is proportional to the amplitude of the modulating signal, which is controlled by the FMVOLUME parameter.

In classic FM synthesizers, all operators are synchronized with each other, and their frequencies are related by a fixed **ratio**.

In **SimplexFM** all operator voices are independent of each other. Therefore, ratio is a **derived** value that depends on the current frequencies of the operators.



If two voices play in an octave, then the ratio is 2:1, in a fifth 3:2, in a fourth 4:3 and so on.



If the frequencies of operators smoothly change under the influence of GLIDE, then the ratio also smoothly changes from the initial value to the final value, and one timbre smoothly evolves into another.



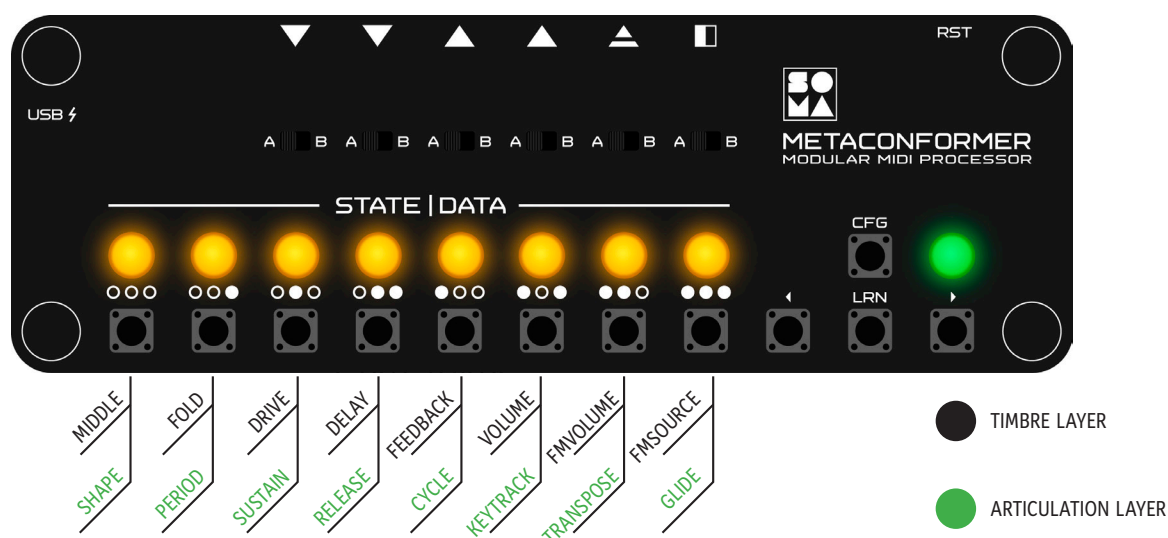
Wavefolder in this perspective can be perceived as a parameter that changes the frequency of the signal in a very unusual manner.

This is simply impossible in a classic FM synthesizer, where the position of harmonics in the timbre is always strictly fixed through the ratio.

## SETTINGS CONTROLS

### Location of parameters

The buttons on the Metaconformer panel are responsible for selecting the current parameter for editing.



The parameters are located on two layers.


**Timbre layer** is responsible for the harmonics of the sound. And the parameters are located in the sequence in which they are found in the audio path.

**The articulation layer** is responsible for the parameters of the volume envelope, key tracking, transposition and glide.


To switch layers, press the **CFG** button. The **articulation** layer corresponds to the green LED glowing, the **timbre** layer corresponds to the LED being turned off.

### Changing values

Editing values is going from your MIDI keyboard or MIDI controller/sequencer. It must be equipped with a mod wheel or knob that is configured to send CC1 (which corresponds to the modulation wheel).

 Make sure that your controller is sending CC1 or mod wheel. For example, MIDI tracks in Elektron devices can write on the screen that they are sending CC1, but by this it means Knob1. This can lead to confusion.

On your MIDI keyboard/controller, go to channel 5, which by default is voice1 channel.

 If something goes wrong, reset your voice parameters (**see initializing voice parameters**) and return to this place.

Editing parameters always consists of 3 simple steps.

1. To select an parameter, switch to the desired layer with the CFG button.
2. Click on one of the buttons 1-8 to select an parameter on the layer.
3. The mod wheel (CC1) of your keyboard controls the voice parameter value.



Be sure to read the sections **Indication** and **Discontinuities avoiding**. They provide additional information that is necessary for management.



You can manage parameters directly through CC messages. See **parameter summary table**.

## DISCONTINUITIES AVOIDING

In **SimplexFM** the control system avoids value discontinuities. For this, the outermost orange LEDs are used – the first and eighth.



Let's say you set MIDDLE to 120 by rotating CC1. Next, you go to the FOLD parameter and set it by rotating the same CC1 to the value of 30. Now, let's return to the MIDDLE parameter. It has a value of 120, but the slightest rotation of CC1 will set it to a value of about 30. There will be a sudden change in value (discontinuity) in the sound, which sounds harsh and attracts unnecessary attention. **SimplexFM** allows you to avoid this and always change the parameters of voices in a smooth way.

After selecting a parameter, the device does not immediately begin to change the value. **SimplexFM** waits until you return CC1 to the previous value of this parameter. In our example, we need to move it from the value 30 to 120.

In order for the user to understand which direction to rotate the handle, the first and eighth LEDs are activated.

The eighth LED lights up when CC1 needs to be turned towards higher values.

The first LED lights up when CC1 needs to be turned towards lower values.

As long as one of these two LEDs is lit, CC1 does not change the sound.

Once you reach the previous value, the 1<sup>st</sup> or 8<sup>th</sup> LEDs will turn off and CC1 will begin to **change** the sound.



— you need to turn CC1 down



— you need to turn CC1 up



— CC1 returned the previous value of the parameter and can now change it without breaks



● — "don't care state" means led state doesn't matter

LEDs 2-7 show the current value, which depends on the scale type.



This control type only works in MONO/STACK **play mode**.

In POLY mode, this method had to be abandoned, since in it several voices can be controlled from one MIDI channel. Although this method will give a smooth sound, it will create confusion, which is much worse than jumps in the sound.

## INDICATION

The **Green** LED only indicates the parameter layer you are on.

**Orange** LEDs show currently relevant information.

If you activate voices with MIDI notes, LEDs 1-4 brightness indicate the volume envelope value of the corresponding voice. The eighth LED shows the current **play mode**.

If CC messages are received by the device, the indicator displays the value on a rough scale of six LEDs. LEDs 2-7.

**SimplexFM** has four scale types.

- a rising scale, when the synthesizer parameter increases with increasing MIDI value. For example, DRIVE

●●●●●●●● — CC1 = 0  
 ●●●●●●●● — CC1 = 64  
 ●●●●●●●● — CC1 = 106  
 ●●●●●●●● — CC1 = 127

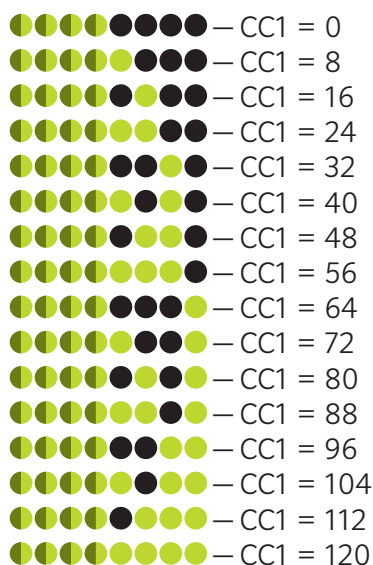
- falling scale, when the synthesizer parameter decreases as the MIDI value increases. For example, the length of the DELAY delay line.

●●●●●●●● — CC1 = 0  
 ●●●●●●●● — CC1 = 12  
 ●●●●●●●● — CC1 = 64  
 ●●●●●●●● — CC1 = 127

- a bidirectional scale that shows the value relative to the middle of the MIDI range. For example, MIDDLE.

●●●●●●●● — CC1 = 0  
 ●●●●●●●● — CC1 = 12  
 ●●●●●●●● — CC1 = 62  
 ●●●●●●●● — CC1 = 63 (VALUE = 0)  
 ●●●●●●●● — CC1 = 64  
 ●●●●●●●● — CC1 = 127

- a switch scale that shows the configuration of the FM Mixer switches. Used only with the FMSOURCE parameter



The first four LEDs show the brightness value of the volume envelope of the corresponding voice

Details about the parameters are presented in the **parameters summary table**.

## SAVING SETTINGS

The values of all parameters are saved to memory automatically. After a restart, the synthesizer restores its state.

### INITIALIZING PARAMETERS

If during your experiments you find yourself in a place where you don't know how to get out, you can reset all voice settings. For this:

1. Press and hold the < (**PREV**) button + 1-4 button of the desired voice for 3 seconds.
2. Restart your device by pressing RST button. All settings for the selected voice will be reset.

MIDI voice channels are set accordingly:

	Main channel	Additional channel
Voice 1	5	5
Voice 2	6	6
Voice 3	7	7
Voice 4	8	8



## MIDI CHANNELS


Each of the 4 **SimplexFM** tracks has a **main** MIDI channel.

MIDI notes from this channel control:

- volume envelope state,
- oscillator frequency,
- delay length of the comb filter if KEYTRACK is turned on.

MIDI CC from this channel will change the value of the selected track parameter.

Also, each of the 4 **SimplexFM** tracks has an **additional** MIDI channel for simultaneous activation of several voices at once.

 Set up **additional** channels of multiple voices on the same MIDI channel, so that notes from that channel trigger multiple voices at once, as if they were the operators of a classic FM synthesizer. Or simply to create a richer, more complex sound.

### SETTING UP MIDI CHANNELS

You can change the MIDI channels of the voices to suit your setup.


This happens by MIDI learn. You send a note to the synthesizer. It sets the voice channel to the same as the note channel. Below is the detailed procedure.

To set the main MIDI channel

1. Press **CFG** to turn the green LED on. You are on the articulation parameters layer.
2. Go to the desired MIDI keyboard channel that you want to assign to one of the tracks.
3. Hold down button 7 on the instrument panel, which is responsible for setting the main MIDI channel.
4. Select one of the voices by pressing and holding one of the buttons 1-4.
5. Press any note on the MIDI keyboard so that the synthesizer saves the channel number of the received note. And release the Metaconformer buttons.
6. Setup is complete.

The additional MIDI channel is configured in the same way.

1. Press **cfg** so that the green LED lights up. You are on the articulation parameters layer.
2. Go to the desired MIDI keyboard channel that you want to assign as an additional channel to one of the tracks.
3. Hold down button 6 on the instrument panel, which is responsible for setting the main MIDI channel.
4. Select one of the voices by pressing and holding one of the buttons 1-4.
5. Press any note on the MIDI keyboard so that the synthesizer remembers the channel number of the received note. And release the Metaconformer buttons.
6. Setup is complete.

 To avoid confusion, it is better to choose an additional channel that is different from the main MIDI voice channels. If you have configured voices on channels 1, 2, 3, 4, then it is better to choose additional ones 5, 6, 7 and so on.

## PLAY MODES

By changing MIDI channels on your MIDI keyboard/controller, you can move from one track to another and control all four **SimplexFM** tracks independently. It's like they're 4 different synthesizers!

But the main feature of **SimplexFM** is FM synthesis. And in order for it to appear, you need at least a couple of voices interacting with each other.

Therefore, **SimplexFM** provides play modes where several voices can be activated from one MIDI channel.

They are changed by pressing **lrrn**.

●●●●●●●● — MONO/STACK — the eighth LED is not lit

●●●●●●●● — POLY — the eighth LED is on



The brightness of the first four LEDs displays the volume envelope value of the corresponding voice.

## MONO/STACK

This is the default mode at startup.

Each voice is a separate synthesizer. Control of parameters and envelope occurs only from the main MIDI channel of the track.

This mode is suitable for playing using an external multi-track sequencer.

Moreover, the envelope can also be triggered from an additional MIDI channel. If two or more tracks have the same additional MIDI channel value, then a note from that channel will trigger all of those tracks.



MIDI CC messages from an additional MIDI channel are ignored and do not change track settings

## POLY

All 4 tracks are combined into a 4-voice multitimbral synthesizer.

In this mode, voice parameters are also controlled from the main MIDI channel of the track.

But the envelope activation occurs only from the main MIDI channel of voice1.



MIDI notes from other MIDI channels are ignored and cannot trigger the voice envelope.

Notes coming from this MIDI channel are distributed across **SimplexFM** tracks. When all voices are filled, the algorithm replaces the oldest note.

0	0	0	0	all voices are empty
1	0	0	0	we get note 1
1	2	0	0	we get note 2
1	2	3	0	we get note 3
1	2	3	4	we get note 4
5	2	3	4	we get note 5, replace note 1
5	2	0	4	note 3 released, track 3 free
5	2	6	4	we get note 6
				and so on...



If CC1 comes from the MIDI channel of voice1, then the parameters of all voices whose envelope is currently being held will be set.

## SUMMARY TABLE OF PARAMETERS

Parameter	Scale	CC	Description
MIDDLE	Bidirectional	10	Asymmetry of the triangular oscillator
FOLD	Falling	11	Wavefolder offset level
DRIVE	Rising	7	Overdrive level
DELAY	Falling	32	Comb filter delay length
FEEDBACK	Bidirectional	13	Comb filter feedback
VOLUME	Bidirectional	16	Volume
FMVOLUME	Bidirectional	34	FM intensity
FMSOURCE	Switch	14	Selecting FM source tracks
SHAPE	Bidirectional	9	Attack-Decay asymmetry
PERIOD	Rising	8	Period of the Attack-Decay stages
SUSTAIN	Rising	19	Sustain Level
RELEASE	Rising	35	Release Time
CYCLE	Bidirectional	27	Enable/disable cyclic envelope mode
KEYTRACK	Bidirectional	56	Comb filter keytrack on/off
TRANSPOSE	Bidirectional	23	Oscillator transposition
GLIDE	Rising	15	Oscillator frequency changing speed

## MASTERING THE INSTRUMENT

It is recommended to split the mastering of **SimplexFM** into several stages. Moreover, you should move to the next stage only when you feel confident in the current one.

### Step 0.

1. Install the firmware on the Metaconformer.
2. Just in case, initialize all voices.
3. Connect a MIDI controller. Make sure you know how to change MIDI channels. By default, **SimplexFM** voices are assigned to channels 5, 6, 7, 8.
4. Make sure you can play each voice separately. Immediately after startup, all voices sound the same, but different LEDs on the panel should light up.
5. Use modwheel to change parameters. If you are using a MIDI controller rather than a keyboard, set one of the knobs to CC1
6. After starting, you will change the first parameter on the timbre layer — MIDDLE. Make sure you can make the sound brighter. But only for the current track. The other tracks should remain unchanged.

### Step 1.



Do not change the FMVOLUME and FMSOURCE parameters yet, Step 2 will be devoted to them

1. Learn to make different timbres. Realize that these are 4 synths in one box, not one polyphonic one.
2. Get used to the parameters and their location.
3. Understand the envelope structure. Listen to how it behaves in cyclic mode.
4. If you don't understand how to return to the original state, initialize the voice parameters. And start again.
5. Listen to how the comb filter sounds. Compare the sound with the key track turned on and off.
6. Once you become confident in creating voices, you can switch to polyphonic mode to play voices from one MIDI channel at a time.
7. Learn to customize voices from polyphonic mode.

## Step 2.

The power of **SimplexFM** is in FM synthesis. Let's consider a recipe for a deep and sticky sound, from which you can build on and then continue your journey on your own.

1. Initialize voice1 and voice2 (or better yet, the others too, so that everything goes according to plan).
2. Set voice2 FMSOURCE to 1000. This means voice1 will modulate it. FMSOURCE of other voices should be 0000
3. Set SimplexFM to polyphonic mode. Make sure you can play Voice1 and Voice2.
4. Set both voices a soft and smooth volume envelope.
5. To start fm, you need to raise the FMVOLUME of voice1. Select FMVOLUME in the panel and change the value.
6. Play voice1 and voice2, increase FMVOLUME until you hear the frequency modulation of voice2 by the sound of voice1.
7. Listen to the result in different octaves. You can use TRANSDPOSE to separate the voices by frequency.
8. Add echo to voice1 and voice2. Make a long DELAY delay in the comb filter and increase the FEEDBACK value.
9. Change the GLIDE value of Voice1 and Voice2.
10. If you already manage to catch the mood, you can create FM feedback. Set Voice1's FMSOURCE to 0100 so that Voice2 modulates it. In this case, the FMSOURCE value of voice2 should remain 1000. Thus, a loop is obtained: voice1 modulates voice2, which modulates voice1.
11. Increase the FMVOLUME of voice2 and listen to the result. The sound should become chaotic. But this chaos goes away along with the volume of voice1 and voice2, since the number of harmonics generated by fm synthesis is proportional to the volume of the sounds involved.
12. Listen to the result in different octaves and at different intervals between the frequencies of voice1 and voice2.
13. Experimenting....



Be prepared that when using feedback between voices, as well as self-modulation, the frequency of the sound will shift in a non-linear manner. During the play, in order for the part to be in key, you should be guided by ear, and not by what notes you enter from the keyboard.



At the same FMVOLUME value, the lower the sound frequency, the higher the degree of distortion.

At high frequencies, fm is noticeable only with large modulus values of the FMVOLUME parameter.

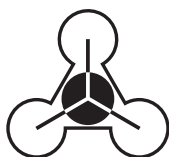
At low frequencies, fm kills periodicity even at moderate absolute values of FMVOLUME.

## CONCLUSION

The name **SimplexFM** comes from the name of a multidimensional polyhedron, each vertex of which is connected by an edge to all other vertices of this polyhedron.

In two-dimensional space, a simplex is a triangle. In three-dimensional space, a simplex is a tetrahedron.

This property quite accurately reflects the frequency modulation capabilities of **SimplexFM**, where each can be connected with each.



When developing a synthesizer, I wanted to create an instrument that would allow to synthesize the smooth evolution of one timbre into another through their superposition.

Not just a parametric glide, when one vector of parameters (or simplified preset) morphs into another. But an evolution that depends on the original sounds.

In this case, the sequence of notes played has a direct effect on the timbre.

That's why **SimplexFM** sounds organismic. It's fun to experiment with.

And finally, **SimplexFM** demonstrates the capabilities of the Metaconformer module itself, which is a completely open system. Its hardware and CPU completely belong to the user. If you know how to program in C, then welcome. Write your own synth and share it with the world!

If you don't know how to program in C, but want to learn how to program for hardware, then Metaconformer will be an excellent starting point for you in this path. All the hardware is already assembled, you just need to learn how to write software.

Start from flashing LEDs, through your own MIDI controller, to a custom polyphonic synth.

## CREDITS

Vadim Minkin — idea, code, music.

Daniil Grigoriev — demo video, music, alpha testing.

Xandr Vasiliev — alpha testing.

Valeriy Zaveryaev — manual design and layout.

Thomas Lundberg — editor, proofreader, alpha testing.

[www.somasynths.com](http://www.somasynths.com)

Vadim Minkin SOMA lab • 2024



